REMARKS/ARGUMENTS

These remarks are offered in response to the Office Action of May 13, 2004 (Office Action). This response is filed after the 3-month shortened statutory period, and as such, a retroactive extension of time is hereby requested. The response is filed concurrently with a Request for Continued Examination (RCE). The Examiner is authorized to charge all appropriate fees to Deposit Account 50-0951.

At page 1 of the Office Action, Applicant's previous amendment to FIG. 1 was objected to under 35 U.S.C. § 132(a) as not being supported by the original disclosure. The previously-filed amendment to FIG. 1 is hereby canceled, as required at page 1 of the Office Action, and the figure thus stands as originally filed.

At pages 1-4 of the Office Action, Claims 7-12 were rejected under 35 U.S.C. § 112, first and second paragraphs. Claim 7 was rejected at page 4 of the Office Action under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 4,313,174 to White (White).

Applicant, as an initial matter, wishes to thank the Examiner for a thorough examination of the application and for discussing by phone various aspects of the invention with Applicant's representative on October 11, 2005. The remarks below address both these aspects and others relevant to the application. Applicant has amended Claims 7-9 and 11. Applicant, moreover, has added Claim 13, directed to a signals-based method aspect of the invention. As discussed herein, the amendments and newly-added claim are fully supported in the Specification.

I. The Claims Are Adequately Supported By Applicant's Disclosure

Claim 7, as amended, is directed to a high-speed scalable multiplier that includes a first signal input for receiving a first signal representing the value of a pre-selected multiplicand and a second signal input for receiving a second signal representing the

value of a pre-selected multiplier. The high-speed scalable multiplier further includes a configuration of electrical circuitry for performing certain arithmetic operations described below. (See, e.g., Specification, p. 13, lines 3-16.)

The Applicant has chosen to designate the configuration of electrical circuitry a "folding multiplier" and the procedure implemented by the circuitry configuration "a folding process." In doing so, Applicant is merely exercising the prerogative afforded all patent applicants to be his own lexicographer, *See, e.g., Innova/Pure Water, Inc. v. Safari Water Filtration Systems, Inc.*, 381 F.3d 1111, 1116-17 (Fed. Cir. 2004); *see also Invitrogen Corp. v. Biocrest Mfg., L.P.* 327 F.3d 1364, 1367 (Fed. Cir. 2003). Moreover, Applicant respectfully maintains that irrespective of any particular terminology, the Specification clearly delineates all aspects of the invention by describing in concise algebraic terms how the invention operates. The circuitry and the algebraic manipulations effected by the circuitry are such that one of ordinary skill could practice the invention without undue experimentation.

The circuitry of the folding multiplier initially generates a first folding value and a second folding value based upon the respective values of the multiplicand and multiplier. The first folding value is explicitly defined in the specification: it is equal to an average of the multiplicand. (See Specification, p. 8, line 19- p.9, line 2.) It is, therefore, computed by taking the sum of the multiplicand and the multiplier and dividing the sum by two. (Specification equation at p. 9, line 1; see also FIG. 1, block 106.) The second folding value is also explicitly defined: one half the difference between the multiplicand and the multiplier. The second folding value is thus generated by taking the difference between the multiplicand and the multiplier and dividing the difference by two. (See Specification, p. 9, lines 2-4; see also FIG. 1, block 106.)

The folding multiplier subsequently generates a first square by squaring the difference between the first folding value and a fractional portion of a first scaling factor. The first scaling factor is equal to (a) one times a predetermined full scale value, if the

first folding value is greater than one half a predetermined full scale value; and (b) zero if the first folding value is less than or equal to one half the full scale value. (See especially FIG. 1, blocks 108, 110, and 112.)

The term "full scale" is explicitly recited in block 108 of FIG. 1, which describes the following decision rule:

> "Is P [the first folding value] > one-half full scale (F)"? If so, then the scaling factor is "K=1*F" (Block 110); otherwise, the scaling factor is "K=0" (Block 112).

Thus, "F," designated parenthetically but nonetheless explicitly in the figure, is stated to be the "full scale" value that is then used to determine the first scaling factor, K. Accordingly, the scaling factor K is either the full scale (i.e., 1*F=F) or zero (i.e., 0*F=0) depending on the result in decision block 108 of FIG. 1.

Although "F" later, in blocks 120 and 122, takes on a different value, this is not misleading to one of ordinary skill in the art. The seeming contradictory use of variables in assignment statements is well known in the art. The oft-cited example is the seemingly nonsensical assignment statement x:=x+1 that is used pervasively in the art. Applicant respectfully maintains that the different use of "F" in the respective blocks of FIG. 1 is not confusing to those of ordinary skill in the art.

The terminology "full scale," moreover, is a commonly employed term in the art. As to this point, Applicant respectfully reminds the Examiner of the earlier-noted phone discussion with Applicant's representative in which the general use of the term was noted. Lest there be any doubt, however, Applicant respectfully submits a sample of current literature, attached hereto under the heading APPENDIX, drawn from a range of fields and applications in which the term is used. It will be noted that in many other references the term is used with little or no comment as to its meaning, suggesting again that the term is so pervasive as to require no additional comment.

The term full scale is understood, moreover, regardless of the base number system to which it is applied. In the context of a base-ten system, for example, the full scale of the integers 0 through 9 is 10. In the context of base-two, applicable in the context of digital circuitry, the full scale of four-bit representations is $(1111)_2$.

The same procedure performed in generating the first square is performed by the folding multiplier for generating a second square: squaring the difference between the second folding value and a fractional portion of a second scaling factor, the scaling factor being based on the full scale. (See Specification, p. 10, lines 5-8; see also FIG. 1, blocks 114, 116, and 118). The second scaling factor is thus equal to (a) one times the full scale value if the second folding value is greater than one half the full scale value; and (b) zero if the second folding value is less than or equal to one half the full scale value.

The folding multiplier further generates a first product by multiplying the first folding value times the first scaling factor, a second product by multiplying the second folding value times the second scaling factor, a third square by squaring the fractional portion of the first scaling factor, and a fourth square by squaring the fractional portion of the second scaling factor. (See Specification, p. 10, lines 9-14; see also FIG. 1, block 120)

Lastly, the folding multiplier, generates a sum of the first square, the first product, and the fourth square. The folding multiplier then subtracts the second square, the second product, and the third square from the sum. The result is termed a "folded product." (See Specification, p. 10, lines 14-17; see also FIG. 1, block 122.) The result is the product of the originally supplied multiplicand and multiplier.

Although the result is termed a "folded product," Applicant respectfully submits that the particular term given the result is not the only thing that describes the invention. Instead, the invention is also defined by the precise algebraic manipulations performed by electronic circuitry. The circuitry is defined as a "folding multiplier" for producing the "folded product" through a set of procedures or functions termed a "folding process."

These, as already noted, are only terms chosen by the Applicant acting as his own lexicographer. Applicant, however, again respectfully submits that the terms alone do not describe the invention. Rather, the invention is further defined by the signal processing that is performed by the circuitry that in performing the functions explicitly defined and described yields the desired result.

Applicant respectfully submits that it can not be said that the invention is not described where the invention is described in the precise and rigorous language of algebra. Algebra has long been the language used in the art to describe various circuit function. In the present context, Applicant respectfully maintains that anyone of ordinary skill would easily glean from the description of the physical and algebraic aspects of the folding multiplier and related procedures contained in the Specification how to carryout the invention without undue experimentation. Applicant respectfully maintains that no more is required of any applicant under 35 U.S.C. § 112.

II. The Claims Define Over The Prior Art

As noted above, independent Claim 7 was rejected as being anticipated by White. White is explicitly directed to a digital arithmetic device for obtaining the square, C^2, of a value C. (Col. 3, lines 21-24; see also Abstract.) White obtains the square of a value using plurality of square function ROMs in implementing the arithmetic operation of a square-by-parts. (Col. 2, lines 10-29.)

One fundamental difference between White and Applicant's invention, therefore, is that, in seeking the square of a single value, White does not provide for the multiplication of a multiplicand by a multiplier, as recited in each of the amended independent claims and newly-added Claim 13. Although the computations effected by White involve two distinct "component values, A and B," neither component value represents a multiplicand or a multiplier. Instead, A corresponds to a value "contributed

by" the most significant bits (MSB) of *C* while *B* corresponds to the value "contributed by" the least significant bits of the same value, namely *C*.

This fundamental difference yields yet another fundamental difference between White and Applicant's invention in terms of how each performs its arithmetic calculations. Applicant's invention generates first and second folding values, each operation explicitly defined in the Specification and illustrated in the accompanying drawings. (See, Specification, p. 8, line 19 – p. 9, line 4; see also FIG. 1) The first folding value is one half times the sum of the values of the multiplicand and multiplier, as recited in each of the amended independent claims as well as newly-added Claim 13. The second folding value is one half times the difference between the values of a multiplicand and a multiplier, as also recited in the amended independent claims and newly-added Claim 13.

There is no comparable computation involving a multiplicand and multiplier performed by White. Indeed, White explicitly states that "there is no addition" performed in calculating C^2. See (Col. 4, lines 52-68.) This inevitably follows since White does not, in fact, deal with two distinct values, but only separate portions – the most significant and least significant bits – of the same input value, *C*. (Col. 4, lines 66-68) As White further states, "bits are concatenated . . . but there is *no actual addition*." (Col 4, line 68 – Col. 5, line 2.)

Moreover, even if this fundamental difference is ignored so that *A* and *B* in White are treated as distinct values rather than different bit-positions of the same value, nevertheless White provides is no computation that yields first and second folding values as explicitly defined in the Specification and recited in the amended independent claims as well as newly-added Claim 13. The scaled product, *4AB*, generated by White involves neither an average of two values nor one-half the difference between two values. (See Col. 5, lines 24-40, especially equations (8) through (10).

Moreover, White no where teaches, either explicitly or inherently, the use of a scaling factor based on a full scale value, as recited in the amended independent claims and newly-added Claim 13. (See Specification, FIG. 1, blocks 108-116.) The scaling factor of White represents "gain scaling," alternatively being 2^N and 2^{N-2}, depending on the plurality of ROMs used. (See Col. 3, line 60- Col. 4, line 5; Col. 5, lines 38-62.) White, however, does not expressly or inherently teach the use of two different scaling factors, *K* and *L*, each equal to one times a full scale or zero, depending on whether corresponding folding values, *P* and *Q*, respectively, are greater than the full scale. (See Specification, Blocks 108 and 114.)

White, furthermore, does no explicitly or inherently teach generating a first square by squaring the difference between the first folding value and a fractional portion of a first scaling factor, such that the first scaling factor is equal to (a) one times a predetermined full scale value, if the first folding value is greater than one half the full scale value, and (b) zero otherwise, as recited in the amended independent claims and newly-added Claim 13. Likewise, White does not explicitly or inherently teach generating a second square by squaring the difference between the second folding value and a fractional portion of a second scaling factor, whereby the second scaling factor is equal to (a) one times the full scale value, if the second folding value is greater than one half the full scale value, and (b) zero otherwise, as also recited in the amended independent claims and newly-added Claim 13. Indeed, White no where expressly or inherently teaches the subtraction of any type of scaling factor from any other value.

It follows that any product of terms in White is determined in a manner entirely distinct from the product determined with Applicant's invention. As recited in each of the independent claims, Applicant's invention determines a product, defined as a folded product, by summing a first square, a first product, and a fourth square, and then subtracting from the sum a second square, a second product, and a third square. White provides no comparable computation in computing the square of a value, *C*. Indeed, as

already noted, White no where discloses computing a first or second square, a first or second product, or a third or forth square. Accordingly, White can not compute a product based on the sum and differences of such terms, as recited in each of the amended independent Claims as well as newly-added Claim 13.

Applicants respectfully maintain that, whereas White fails to expressly or inherently teach the features recited in amended independent Claims 7, 9, 11, and 13, the claims define over the prior art. Applicants further respectfully assert that whereas each of the remaining claims depends from one of the amended independent claims, these claims likewise define over the prior art.

CONCLUSION

Applicant believes that this application is now in full condition for allowance, which action is respectfully requested. Applicant requests that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

Date: November 14, 2005

Gregory A. Nelson, Registration No. 30,577
Richard A. Hinson, Registration No. 47,652
AKERMAN SENTERFITT
Customer No. 30448
Post Office Box 3188
West Palm Beach, FL 33402-3188
Telephone: (561) 653-5000

APPENDIX

TAB A: Ernest O. Doebelin, MEASUREMENT SYSTEMS APPLICATION AND DESIGN (McGraw-Hill, 1989), pp. 826-830.

TAB B: Iotech, *Handbook*, available at <http://www.iotech.com/handbook/chapt_2.html>, excerpt from Chapter 2, Analog-to-Digital Conversion.

TAB C: National Semiconductor, *D/A Converters: Definition of Terms* (May 1999).

TAB D: Bonnie C. Baker, Microchip Technology, Inc., AN753: DIGITAL CODING SCHEMES FOR MIXED SIGNAL COMMUNICATION, (Microchip, 2001), pp. 1-2.

TAB E: Canberra, *Analog-to-Digital Converter* (2002).

TAB F: James Bryant, Ask The Applications Engineer: Analog Dialogue 22-2 (Analog Devices, 1997), pp. 1-3.

TAB G: Oli Josefsson, Ask The Applications Engineer: Analog Dialog 28-2 (Analog Devices, 1997, p. 37-38.

10.12 ANALOG-TO-DIGITAL AND DIGITAL-TO-ANALOG CONVERTERS; SAMPLE/HOLD AMPLIFIERS

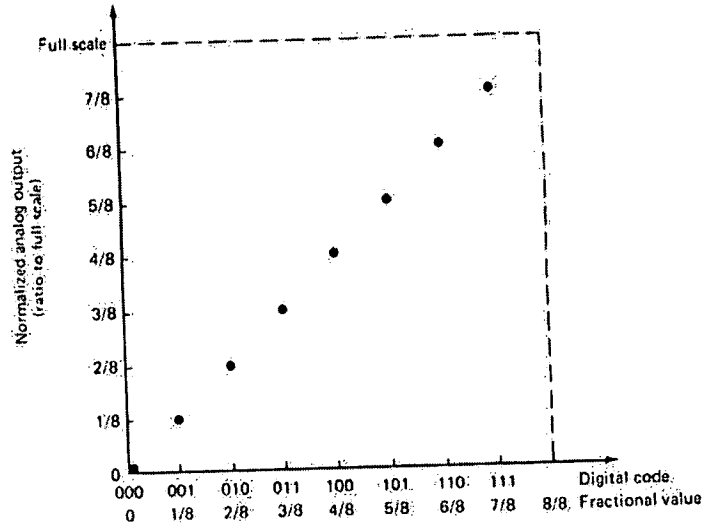
Because most sensors have analog output while much data processing is accomplished with digital computers, devices for conversion between these two realms obviously play an important role. For motion inputs, the shaft-angle encoders of Chap. 4 provide analog-to-digital (A/D) conversion; here we concentrate on all-electronic devices whose analog input or output is a voltage.

We begin with digital-to-analog (D/A) converters since they are used as components in some A/D converters. The most fundamental property of any D/A or A/D converter is the number of bits for which it is designed, since this is a

Binary Bits (n)	(2^n)	Equivalent percent or fraction of range of least-significant bit*		Residual $1-\sum_1^n\left(\frac{1}{2^n}\right)$
		Percent	ppm	
1	2	50.0	500 000.	0.5
2	4	25.	250 000.	0.25
3	8	12.5	125 000.	0.125
4	16	6.25	62 500.	0.062 5
5	32	3.125	31 250.	0.031 25
6	64	1.562 5	15 625.	0.015 625
7	128	0.781 25	7 812.5	0.007 812 5
8	256	0.390 625	3 906.25	0.003 906 25
9	512	0.195 313	1 953.13	0.001 953 13
10	1 024	0.097 656	976.56	0.000 976 56
11	2 048	0.048 828	488.28	0.000 488 28
12	4 096	0.024 414	244.14	0.000 244 14
13	8 192	0.012 207	122.07	0.000 122 07
14	16 384	0.006 104	61.04	0.000 061 04
15	32 768	0.003 052	30.52	0.000 030 52
16	65 536	0.001 526	15.26	0.000 015 26
17	131 072	0.000 763	7.63	0.000 007 63
18	262 144	0.000 381	3.81	0.000 003 81
19	524 288	0.000 191	1.91	0.000 001 91
20	1 048 576	0.000 095	0.95	0.000 000 95
21	2 097 152	0.000 048	0.48	0.000 000 48
22	4 194 304	0.000 024	0.24	0.000 000 24
23	8 388 608	0.000 012	0.12	0.000 000 12
24	16 777 216	0.000 006	0.06	0.000 000 06

* May be limited by noise and other uncertainties in actual circuit.

Figure 10.57 Resolution of A/D and D/A converters.



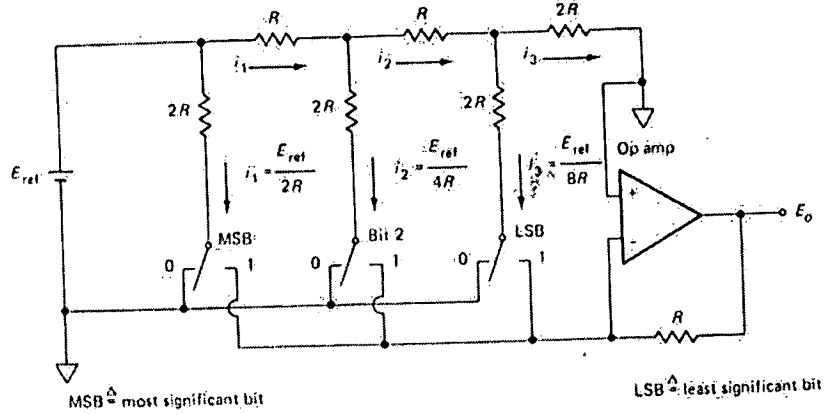


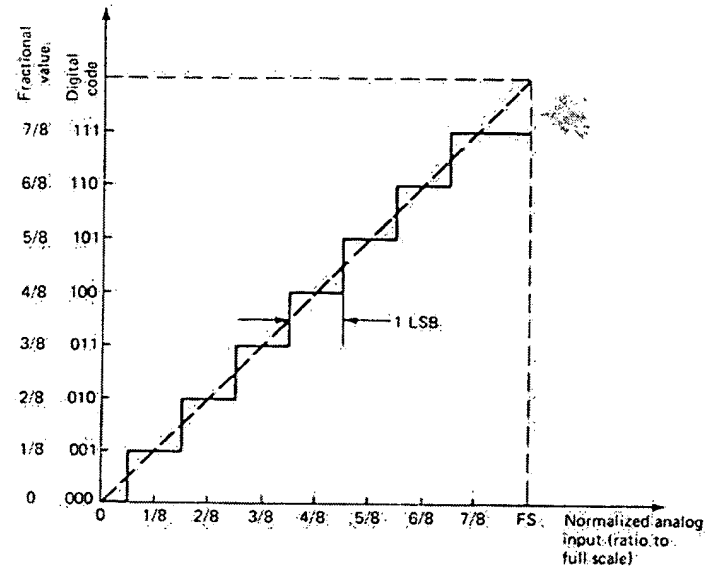
Figure 10.58 Digital-to-analog conversion (3 bits).

basic limit on resolution (see Fig. 10.57)[1]. Units with 8 to 12 bits are most common; however, resolution to about 18 bits is available, but special care must be taken in the application[2] (the least significant bit represents only 38 μV). Most D/A converters utilize a principle similar to that of Fig. 10.58, the so-called R-2R

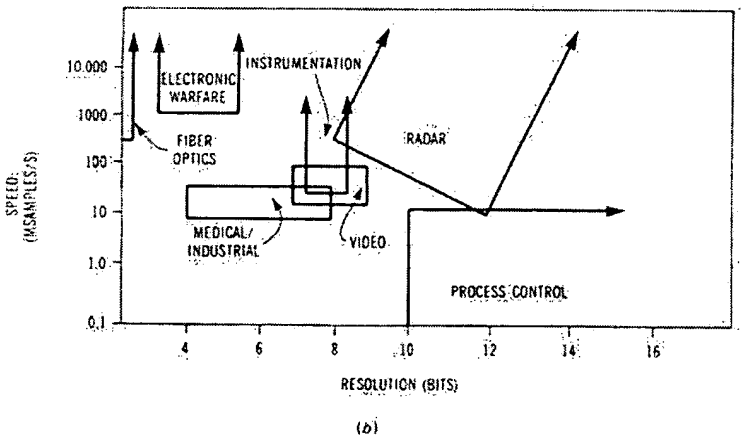
[1] Design Engineers' Handbook and Selection Guide, A/D and D/A Converter Modules, BR-1021, Analogic Corp., Wakefield, Mass., 1980.

[2] Designer's Guide to High Resolution Products, Analog Devices, Norwood, Mass., 1981; "Analog-Digital Conversion Handbook," 3d ed., Analog Devices, Norwood, Mass., 1986.

ladder network. Basic to the accuracy of such devices is the stability of the reference voltage and resistor values. Even with the best components, the highest-resolution converters need periodic calibration to remain within specification. To understand the operation of the 3-bit D/A circuit shown, note that the switches (called *current-steering switches*) are connected to "ground" irrespective of whether they are "on" (1) or "off" (0), because the op-amp negative input is a virtual



(*a*)



(*b*)

Figure 10.59 Analog-to-digital conversion.

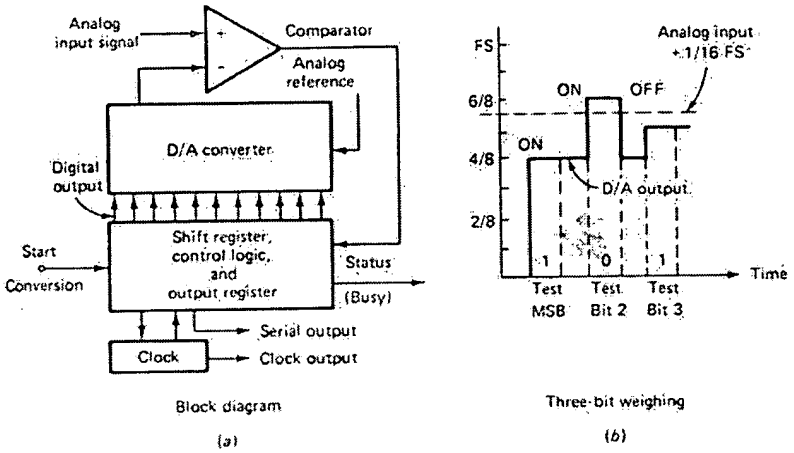


Figure 10.60 Successive-approximation A/D converter.

ground. So we can redraw the circuit in this way and quickly see why the currents are as labeled. The op-amp merely sums the currents "steered" to it by the switches and produces an output voltage proportional to this sum.

A wide variety of D/A converters with a range of cost/performance and special features are available from many manufacturers: bipolar (± 10 V) outputs (the unit of Fig. 10.58 is unipolar, 0 to 10 V), current outputs (such as the process control 4- to 20-mA standard), multiplying types in which E_{ref} can be a dynamic variable, digitally buffered (data-bus-compatible) types for easy microprocessor interfacing, etc. Manufacturers' catalogs and handbooks provide comprehensive listings and discussions of specifications. In addition to resolution (mentioned earlier), here we mention only that settling times [within $\frac{1}{2}$ least-significant bit (LSB)] for full-scale input are in the range of 3 to 30 μs for "ordinary" units, while about 10 ns can be achieved by 8-bit "video" converters.

Let's turn now to A/D converters.[1] The majority of practical applications employ one of two different principles: successive approximation or dual slope (integrating). The dual-slope type is superior in most respects, except speed; however, its speed limitation rules out many applications, so successive-approximation types are very common. Figure 10.59*a* shows the ideal behavior of a 3-bit A/D converter. Note the inherent quantization uncertainty of $\pm\frac{1}{2}$ LSB. A successive-approximation A/D converter operates according to the block diagram of Fig. 10.60.[2] When the "start conversion" command is applied, the D/A converter (which is built into the A/D device) outputs the most significant bit

[1] B. M. Gordon, Linear Electronic Analog/Digital Conversion Architectures, Their Origins, Parameters, Limitations and Applications, *IEEE Trans.*, vol. CAS-25, no. 7, July 1978.

[2] D. H. Sheingold (ed), "Analog-Digital Conversion Notes," Analog Devices, Norwood, Mass., 1977.

(MSB) for comparison with the analog input. If the input is greater than the MSB, the MSB remains on (1 in the output register) and the next smaller bit is tried; if the input is less than the MSB, the MSB is turned off (0 in the output register) and the next smaller bit is tried. If the second bit does not add enough weight to exceed the input, it is left on and the third bit is tried. If the second bit "tips the scales" too far, it is turned off and the third bit is tried. This process continues, in order of descending bit weight, until the last bit has been tried, at which point the status line changes state to indicate that the contents of the output register now constitute a valid conversion. (To force the transitions to occur at the ideal $\frac{1}{2}$ LSB points of Fig. 10.59, the comparator is biased by $\frac{1}{2}$ LSB.) This type of converter cannot tolerate much change in the analog input during the conversion process, so a sample/hold device must be used ahead of the converter if fast-changing analog signals are to be converted accurately. The status output of the converter can be employed to release the sample/hold from its hold mode at the end of conversion. Converters of this type are available to about 16 bits (conversion time ≈ 30 μs); 8-bit units can be as fast as 1 or 2 μs.

Several variations of the dual-slope principle are utilized; Fig. 10.61[1] shows a basic implementation. Conversion here is indirect since the analog input is first converted to a time interval which is then digitized by using a counter. The analog input V_{in} is applied to an integrator, and a counter (counting clock pulses) is started at the same time. After a preset number of counts (and thus a fixed time T), the input is disconnected and a reference voltage of opposite polarity is applied to the integrator. At this switching instant, integrator output is proportional to the average value of V_{in} over time interval T. The integral of V_{ref} is an opposite-going ramp of slope $V_{ref}/(RC)$. The counter, reset to zero at time T, now counts until the integrator output crosses zero, Δt seconds later. Therefore Δt (and thus counter output) will be proportional to the average value of V_{in} over time interval T. In Fig. 10.61, V_{in} has been offset by V_{ref} and divided by 2, which allows a bipolar analog input to produce an offset binary output suitable as input for computer systems.

Dual-slope converters have a number of advantages. Accuracy is unaffected by capacitor value or clock frequency, since these effect the up slope and down ramp equally. The integrating effect rejects high-frequency noise and averages changes in V_{in} during the integration period T. By choosing T as an integral multiple of the most prevalent noise signal's period (say, $T = \frac{1}{60}$ s for 60-Hz noise), theoretically perfect noise rejection at 60, 120, 180 Hz, etc., is obtained. Of course, this choice leads to the main disadvantage of dual-slope A/D converters: the slow conversion rate, usually less than 30 per second. A final advantage, widely implemented, is the easy inclusion of automatic-zero capability. Here, before each measurement cycle the input is short-circuited; thus any output produced by the integration process over T must be zero-drift error, which is subsequently subtracted from the next reading, to give continuous zero correction. (Successive-approximation A/D devices do not use automatic zero because of the speed

[1] Ibid.

dB below input signal average

-2

-4

Figure

penalt
Conve
per se
tion.

1. http://www.iotech.com/handbook/chapt_2.html
This is Chapter 2 from a book of a company called iotech. Book details are at the bottom of this web page

Chapter 2
Analog to Digital Conversion

ADC TYPES
Analog-to-Digital Converters (ADCs) transform an analog voltage to a binary number (a series of 1's and 0's), and then eventually to a digital number (base 10) for reading on a meter, monitor, or chart. The number of binary digits (bits) that represents the digital number determines the ADC resolution. However, the digital number is only an approximation of the true value of the analog voltage at a particular instant because the voltage can only be represented (digitally) in discrete steps. How closely the digital number approximates the analog value also depends on the ADC resolution.

A mathematical relationship conveniently shows how the number of bits an ADC handles determines its specific theoretical resolution: An n-bit ADC has a resolution of one part in 2^n. For example, a 12-bit ADC has a resolution of one part in 4,096, where 2^{12} = 4,096. Thus, a 12-bit ADC with a maximum input of 10 Vdc can resolve the measurement into 10 Vdc/4096 = 0.00244 Vdc = 2.44 mV. Similarly, for the same 0 to 10-Vdc range, a 16-bit ADC resolution is $10/2^{16}$ = 10/65,536 = 0.153 mV. The resolution is usually specified with respect to the full-range reading of the ADC, not with respect to the measured value at any particular instant.

Successive-Approximation ADCs
A successive-approximation converter, Figure 2.01, is composed of a digital-to-analog converter (DAC), a single comparator, and some control logic and registers. When the analog voltage to be measured is present at the input to the comparator, the system control logic initially sets all bits to zero. Then the DAC's ***most significant bit (MSB) is set to 1, which forces the DAC output to 1/2 of full scale (in the case of a 10-V full-scale system, the DAC outputs 5.0 V).***
The comparator then compares the analog output of the DAC to the input signal, and if the DAC output is lower than the input signal, (the signal is greater than 1/2 full scale), the MSB remains set at 1. If the DAC output is higher than the input signal, the MSB resets to zero. Next, the second MSB with a weight of 1/4 of full scale turns on (sets to 1) and forces the output of the DAC to either 3/4 full scale (if the MSB remained at 1) or 1/4 full scale (if the MSB reset to zero). The comparator once more compares the DAC output to the input signal and the second bit either remains on (sets to 1) if the DAC output is lower than the input signal, or resets to zero if the DAC output is higher than the input signal. The third MSB is then compared the same way and the process continues in order of descending bit weight until the LSB is compared. At the end of the process, the output register contains the digital code representing the analog input signal.

Successive approximation ADCs are relatively slow because the comparisons run serially, and the ADC must pause at each step to set the DAC and wait for its output to settle. However, conversion rates easily can reach over 1 MHz. Also, 12 and 16-bit successive-approximation ADCs are relatively inexpensive, which accounts for their wide use in many PC-based data acquisition systems.

Voltage-to-Frequency ADCs
Voltage-to-frequency ADCs convert the analog input voltage to a pulse train with the frequency proportional to the amplitude of the input (See Figure 2.02). The pulses are counted over a fixed period to determine the frequency, and the pulse counter output, in turn, represents the digital voltage.
Voltage-to-frequency converters inherently have a high noise rejection characteristic, because the input signal is effectively integrated over the counting interval. Voltage-to-frequency conversion is commonly used to convert slow and noisy signals. Voltage-to-frequency ADCs are also widely used for remote sensing in noisy environments. The input voltage is converted to a frequency at the remote location and the digital pulse train is transmitted over a pair of wires to the counter. This eliminates noise that can be introduced in the transmission lines of an analog signal over a relatively long distance.

National Semiconductor

May 1999

D/A Converters
Definition Of Terms

Differential Nonlinearity: Ideally, any two adjacent digital codes correspond to output analog voltages that are exactly one LSB apart. Differential non-linearity is a measure of the worst case deviation from the ideal 1 LSB step. For example, a DAC with a 1.5 LSB output change for a 1 LSB digital code change exhibits ½ LSB differential non-linearity. Differential non-linearity may be expressed in fractional bits or as a percentage of full scale. A differential non-linearity greater than 1 LSB will lead to a non-monotonic transfer function in a DAC.

Gain Error (Full Scale Error): The difference between the output voltage (or current) with full scale input code and the ideal voltage (or current) that should exist with a full scale input code.

Gain Temperature Coefficient (Full Scale Temperature Coefficient): Change in gain error divided by change in temperature. Usually expressed in parts per million per degree Celsius (ppm/°C).

Integral Nonlinearity (Linearity Error): Worst case deviation from the line between the endpoints (zero and full scale). Can be expressed as a percentage of full scale or in fraction of an LSB.

LSB (Lease-Significant Bit): In a binary coded system this is the bit that carries the smallest value or weight. Its value is the full scale voltage (or current) divided by 2^n, where n is the resolution of the converter.

Monotonicity: A monotonic function has a slope whose sign does not change. A monotonic DAC has an output that changes in the same direction (or remains constant) for each increase in the input code. the converse is true for decreasing codes.

MSB (Most Significant Bit): In a binary coded system this is the bit that has the largest value or weight. Its value is one half of full scale.

Multiplying DAC: In a sense, every DAC is a multiplying DAC since the output voltage (or current) is equal to the reference voltage times a constant determined by the digital input code divided by 2^n (n is the number of bits of resolution). In a two quadrant multiplying DAC the reference voltage or the digital input code can change the output voltage polarity. If both the reference voltage and the digital code change the output voltage polarity, four quadrant multiplication exists.

Offset Error (Zero Error): The output voltage that exists when the input digital code is set to give an ideal output of zero volts. All the digital codes in the transfer curve are offset by the same value. Offset error is usually expressed in LSBs.

Power supply Rejection (Power Supply Sensitivity): The sensitivity of a converter to changes in the dc power supply voltages.

Resolution: the smallest analog increment corresponding to a 1 LSB converter code change. For converters, resolution is normally expressed in bits, where the number of analog levels is equal to 2^n.

Settling Time: The time from a change in input code until a DAC's output signal remains within ±½ LSB (or some other specified tolerance) of the final value.

MICROCHIP

AN753

Digital Coding Schemes for Mixed Signal Communication

Author: *Bonnie C. Baker*
Microchip Technology Inc.

OVERVIEW

An Analog-to-Digital (A/D) converter translates an analog input signal into a discrete digital code. This digital representation of the "real world" signal can be manipulated in the digital domain for the purposes of information processing, computing, data transmission or control system implementation. In any application where a converter is used, it is advantageous to have the code structure complement the microcontroller's operands.

This application note describes the straight binary and binary two's complement code schemes that are outputted by Microchip's Analog-to-Digital (A/D) converters.

All code examples given in this application note are for a 4-bit conversion. The median analog voltages in the tables are the equivalent analog voltages that are at the center of the digital code.

STRAIGHT BINARY CODE

The straight binary code is more accurately called unipolar straight binary. This digital format for an A/D conversion is the simplest to understand. As the name implies, this coding scheme is used only when positive voltages are converted. An example of this type of coding is shown in Table 1.

When this scheme is used to represent a positive analog signal range, the digital code for zero volts is equal to zero (0000 per Table 1). Given an ideal converter with no offset, gain, INL or DNL error, the code transition from 0000 to 0001 occurs at the analog value of:

$$First\ Code\ Transition = \left(0 + \frac{1}{2}LSB\right)$$

$$Second\ Code\ Transition = \left(1LSB + \frac{1}{2}LSB\right)$$

where:

$$LSB = \frac{+FS}{2^n}$$

where:

n is equal to the number of bits in the converter

+FS is equal to the analog full-scale range.

Median Analog Voltage (V)	Digital Code
0.9375 FS ($^{15}/_{16}$ FS)	1111
0.875 FS ($^{14}/_{16}$ FS)	1110
0.8125 FS ($^{13}/_{16}$ FS)	1101
0.75 FS ($^{12}/_{16}$ FS)	1100
0.6875 FS ($^{11}/_{16}$ FS)	1011
0.625 FS ($^{10}/_{16}$ FS)	1010
0.5625 FS ($^{9}/_{16}$ FS)	1001
0.5 FS ($^{8}/_{16}$FS)	1000
0.4375 FS ($^{7}/_{16}$ FS)	0111
0.375 FS ($^{6}/_{16}$ FS)	0110
0.3125 FS ($^{5}/_{16}$ FS)	0101
0.25 FS ($^{4}/_{16}$ FS)	0100
0.1875 FS ($^{3}/_{16}$ FS)	0011
0.125 FS ($^{2}/_{16}$ FS)	0010
0.0625 FS ($^{1}/_{16}$ FS)	0001
0	0000

TABLE 1: *The unipolar straight binary code representation of zero volts is equal to a digital (0000). The analog full-scale minus one LSB digital representation is equal to (1111). With this code, there is no digital representation for analog full-scale.*

The A/D converters from Microchip that produce a straight binary output code are from the MCP320X (12-bit) and the MCP300X (10-bit) families.

These devices can be operated in a single ended, positive voltage input mode or a pseudo-differential input mode, but in both cases the digital output represents a positive input voltage. In the pseudo-differential mode, the IN- input is limited to ±100 mV. This can be used to cancel small noise signals present on both the IN+ and IN- inputs. This provides a means of rejecting noise when the IN- input is used to sense a remote signal ground. The converter will produce digital code that represents the analog input when the IN+ input range is from IN- to (V_{FS} -1 LSB). When the voltage level of IN+ is less than IN-, the resultant code for the family of devices will be still be '0', which does not represent a negative voltage.

BINARY TWO'S COMPLEMENT CODE

In some applications it may be necessary for an ADC to convert negative and positive values. The logic modification that allows this flexibility in the digital output code is to produce the bipolar results called offset binary two's complement. Binary two's complement arithmetic is widely used in microcontrollers, calculators and computers.

Binary two's complement is not as straight forward as the scheme for straight binary. The codes are not continuous from one end to the other due to the discontinuity that occurs at the analog bipolar zero.

The two's complement of a negative binary number is generated by logically complementing all the digits of the positive binary number, hence converting it to the negative binary number counterpart as shown in Table 2. With this coding scheme, the MSB can be considered a sign indicator. When the MSB is a logic '0', a positive value is indicated and when the MSB is a logic '1', a negative value is indicated.

This system is has an odd number of codes and only one zero state. It is also mathematically consistent making it synergistic with signed arithmetic functions.

Median Voltage (V)	Digital Code
0.875 FS ($^7/_8$ FS)	0111
0.75 FS ($^6/_8$ FS)	0110
0.625 FS ($^5/_8$ FS)	0101
0.5 FS ($^4/_8$ FS)	0100
0.375 FS ($^3/_8$ FS)	0011
0.25 FS ($^2/_8$ FS)	0010
0.125 FS ($^1/_8$ FS)	0001
0	0000
-0.125 FS (-$^1/_8$ FS)	1111
-0.25 FS (-$^2/_8$ FS)	1110
-0.375 FS (-$^3/_8$ FS)	1101
-0.5 FS (-$^4/_8$FS)	1100
-0.625 FS (-$^5/_8$ FS)	1011
-0.75 FS (-$^6/_8$ FS)	1010
-0.875 FS (-$^7/_8$ FS)	1001
-1 FS	1000

TABLE 2: *The binary two's complement representation of zero volts is also equal to a digital (0000). The analog positive full-scale minus one LSB digital representation is equal to (0111) and the analog negative full-scale representation is (1000).*

The A/D converters from Microchip that produce a binary two's complement output code are from the TC340X, TC53X, TC7109, TC85 and all I^2C/SMBus thermal sensors families.

These devices are operated in a full-differential input mode. In this mode, the full-scale range of the device is equal to:

$$FS\,range = (IN+_{MAX} - (IN-_{MIN})) + (IN-_{MAX} - (IN+_{MIN}))$$

And the input voltage presented to the converter is equal to:

$$AIN = ((IN+)-(IN-))$$

These converters will produce digital code that represents both negative and positive analog inputs.

CANBERRA

ANALOG-TO-DIGITAL CONVERTER

An analog-to-digital converter (ADC) generates a digital word proportional to the amplitude of an input pulse. In nuclear applications, ADCs are used to digitize the output signals from spectroscopy amplifiers. Since these amplifiers generate output pulses whose amplitudes are directly proportional to the energies of the incident radiation, the ADC can be used with an amplifier and a multichannel analyzer (MCA) to generate energy distributions (spectra) of radioactive samples.

Canberra offers two types of ADCs to fulfill a wide variety of applications: the Wilkinson ADC and the Fixed Dead Time (FDT) ADC.

The Wilkinson ADC

The Wilkinson ADC digitizes a pulse by charging a capacitor to the amplitude of the input signal. The capacitor is then discharged at a constant rate until the voltage ramp returns to a baseline reference. The capacitor discharge time, determined by a crystal controlled clock and register, is directly proportional to the amplitude of the input signal. The number in the register represents the input signal pulse height. This process is indicated schematically in Figure 1.

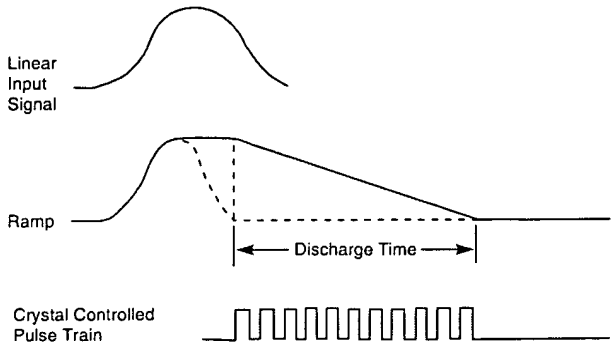


Figure 1 ADC Ramp and Pulse Train

The Fixed Dead Time ADC

The Fixed Dead Time (FDT) ADC, (sometimes called the Fixed Conversion Time ADC), uses successive approximation to digitize its input signal. Unlike the Wilkinson, the Canberra FDT ADC performs a number of comparisons to determine the digital value of the signal.

The successive approximation technique compares the analog input to a binary weighted reference voltage. When a conversion is initiated, the successive approximation register (SAR) activates each bit synchronously with the system clock, and tests each bit's contribution. Bit contributions that do not exceed the analog input are left on, while bit contributions that exceed the analog input are removed when the next lower bit is tested. This process repeats until all bits have been tested. When completed, a digital word or address will represent the analog input. Figure 2 illustrates the Fixed Conversion Time Principle of the successive approximation technique and Figure 3 illustrates the FDT ADC conversion process.

Phone contact information

Benelux/Denmark (32) 2 481 85 30 • Canada 905-660-5373 • Central Europe +43 (0)2230 37000 • France (33) 1 39 48 57 70 • Germany (49) 6142 73820
Japan 81-3-5844-2681 • Russia (7-095) 429-6577 • United Kingdom (44) 1235 838333 • United States (1) 203-238-2351
For other international representative offices, visit our Web Site: *http://www.canberra.com* or contact the Canberra U.S.A. office.

Rev. 8/02 Printed in U.S.A.

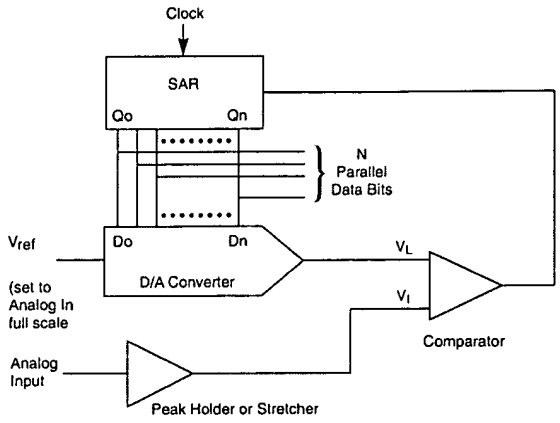


Figure 2 FDT Conversion Time ADC Principle

ADC Characteristics

Specific ADC characteristics relating to the ramp slope, number of channels, linearity, and other features, are listed and described below:

- ADC Resolution
- Conversion Gain
- Conversion Time
- Dead Time and Live Time
- Digital Offset
- Linearity
- Coincidence and Anticoincidence Gating
- Discriminator and ADC Zero Level

The resolution of an ADC is usually defined in terms of channels, and represents the number of discrete voltage increments that the full input pulse height range can be subdivided. Although the full input pulse height range can be any voltage, most nuclear ADCs are designed to be compatible with the output of nuclear spectroscopy amplifiers. Thus, Canberra Wilkinson and FDT ADCs accept direct coupled, positive unipolar or bipolar (positive lobe leading) pulses for PHA with an amplitude of 0 to 10 V dc, and are generally referred to as having an ADC resolution of 4096, 8192 or 16 384 channels.

ADC conversion gain is similar to ADC resolution. However, ADC conversion gain refers to the number of discrete voltage levels into which the ADC's full scale input will be divided for a particular application.

Whereas NaI(Tl) detector applications usually benefit from a conversion gain of 512 or 1024 channels due to their common use in non-multiplet applications. HPGe detectors benefit more from the use of conversion gains of 4096 or 8192 channels due to their common use in sophisticated multi-nuclide applications.

Depending on the particular model purchased, all Canberra ADCs feature a multi-position rotary switch that is used to select conversion gains of 512, 1024, 2048, 4096, 8192 and possibly 16 384 channels (i.e., 8 to 14 bits respectively) for a 10 V input pulse or level.

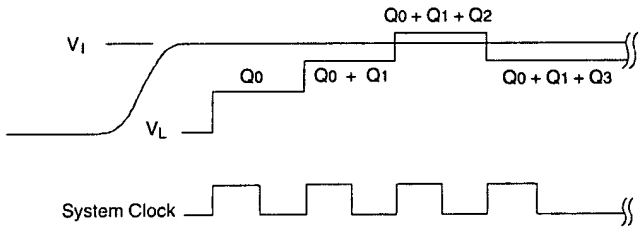


Figure 3 FDT Conversion Process

ADC conversion time refers to the time required to change an input signal from an analog voltage to a digital number. In the case of Wilkinson ADCs, the higher the clock frequency of the Wilkinson ADC, the shorter the ramp time for the same number of channels and the lower the dead time. Dead time refers to the time the ADC is busy processing one pulse and can not accept another. The conversion time for a typical Canberra 100 MHz Wilkinson ADC (Model 8701) is given by the equation:

Wilkinson (T) = 1.5 μs + 0.01(N+X) μs
Where: N = Address Count
X = Effective Digital Offset

The 1.5 μs part of the conversion time follows the ramp generation (see Figure 1). The total conversion time is longer since it includes the input pulse width. This is referred to in Canberra specifications as the linear gate time.

We can see by the equation that the actual conversion time of a Wilkinson ADC is a function of the amplitude of the pulse. However, the conversion time is also a function of the selected full scale conversion gain. For small full scale conversion gains, such as 256, 512 or 1024 channels, the equation yields conversion times of approximately 4.1, 6.6 and 11.4 μs respectively. However, in actual applications, the average conversion time is even faster because statistically it can be proven that for most applications, the average converted channel number (i.e., address) is located at approximately 30% of the full scale conversion gain. Thus, for low to moderate energy applications with PIPS or NaI(Tl) detectors, the Wilkinson ADC is a suitable choice.

In contrast, Fixed Dead Time ADCs feature a fixed conversion time which is dependent upon the amount of comparisons performed, and not the amplitude of the input pulse. For a 8192 channel conversion gain, which actually converts between 0 and 8191 channels, 12 comparisons are required to determine the converted channel number (i.e., address). Since the amount of time it takes to perform the comparisons is known by design, the total amount of fixed conversion time is known. If the user should select conversion gains smaller than the maximum full scale conversion gain of the ADC, the same number of comparisons are always made, however, the least significant bits are thrown away. Thus, the conversion time of a FDT ADC remains constant. The dead time of a FDT ADC is equal to the linear gate time plus the conversion time.

Generally, the Fixed Dead Time ADC finds itself useful in moderate to high energy applications that require full scale conversion gains of 4096 channels or greater. Additionally, the FDT ADC is useful in high count rate applications.

ADC dead time, usually expressed as a percentage, is indicated on an LED bar graph found on the front panel of all Canberra NIM-type ADCs, and/or on the CRT display of Canberra's MCAs. The MCA compensates for dead time by acquiring data for live time periods, rather than clock or real (true) time. Live time is obtained by gating off a clock during the ADC's dead time. All Canberra MCAs may be preset on "live time" or "true (real)

time". To reduce dead time, all Canberra Wilkinson and Fixed Dead Time ADCs include a one-conversion buffer register which allows the ADC to process a subsequent pulse while MCA memory is being incremented.

Digital offset is the capability of subtracting a specified channel number from the converted channel number (or address) before memory is accessed. This is equivalent to using a biased amplifier, but has the advantage of being digital. Digital offset is used frequently in alpha spectroscopy to remove the lower energy ranges and expand a higher energy range across the full memory. Digital offset is also used to collect data at high resolution with a small MCA memory assignment. For example, with an 8192 channel ADC gain and a 1024 channel MCA, a digital offset of 7168 allows data to be acquired for the top eighth of the spectrum.

Linearity of an ADC is given in terms of integral and differential nonlinearity. Integral nonlinearity may be described as the deviation in channels from a linear relationship of pulse height vs. channel number, while differential nonlinearity may be described as the variation in channel width from one channel number to another. Typical values are less than 0.025% (about 1 channel in 4000) for integral nonlinearity and less than 1% for differential nonlinearity.

The ADCs also provide coincidence/anticoincidence gating. If gating is used, most Canberra ADCs require that the gate pulse be at least 250 ns wide. For early coincidence, the gate pulse must occur during the crossing of the input threshold. For late coincidence, the gate pulse must occur during the time that the ADC's linear gate is open (that is, during the ADC acquisition time), between the time the input signal crosses the threshold discriminator to the point where it reaches the peak maximum and falls to 90% of peak height, as shown in Figure 4. Some ADCs allow holding the linear gate open for a longer period of time needed to do coincidence gating, while in other cases the ADC input signals may need to be delayed. The gate pulse must be TTL compatible; a standard SCA output pulse is adequate.

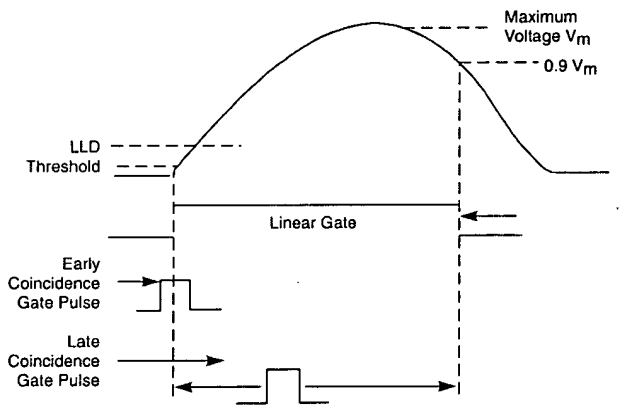


Figure 4 Input Pulse and Coincidence Gate Timing

Slowly varying signals or dc levels may be sampled and digitized to obtain an amplitude distribution curve. When the ADC is set for Sample Voltage Analysis (SVA) mode, a sampling signal at the gate input determines when the signal will be digitized.

Lower and upper level discriminators set an energy window to allow rejection of input signals from the conversion process if their amplitude is outside the selected window range.

The ADC zero level, or analog baseline, is adjustable over a small range in order to allow precise setting of the zero energy intercept.

Ask The Applications Engineer—1

by James Bryant

MULTI TROUBLES

Q. My multiplexed ADC system is misbehaving . . .

A. Before you go any further, have you grounded all unused multiplexer channels?

Q. No. But how did you know?

A. Because the floating terminal is one of the commonest causes of problems in systems containing CMOS multiplexers. Unused MUX inputs and outputs (whether integrated into a multiplexed ADC or part of a self-contained MUX chip) can pick up signals from stray fields and inject them into the device's substrate, turning on spurious substrate devices. Then, even when the unused channel is turned off, the performance of the *on-channel may be* badly degraded (at *the* unlikely extreme, *the* injection may turn on a spurious four-layer device and destroy some chips).

Whenever a MUX is used, all its inputs and outputs must be connected to a potential between its supply rails. The best way to deal with unused channels is to ground them, but they may be connected to a more-convenient potential within the rails. ◘

TROUBLE FROM THE START

Q. To save power, my ADC is powered up only to make a measurement. The system is very accurate in continuous operation, but unpredictable when power is strobed. Why?

A. When an ADC's power is switched on only to perform a conversion, it may misbehave for three reasons: slow reference turn-on, random initial logic states, and system latch-up.

For various reasons—thermal stabilization, capacitance charging, slow starting of regenerative current mirrors using PNP transistors in band-gap references— it is not uncommon for some voltage references to have relatively large errors for many milliseconds after power-up. Such errors in an ADC's external or internal reference during conversion lead to inaccurate results.

At turn-on, a typical ADC's logic will be in a random state; for a conversion triggered at that time, the ADC may not be able to perform correctly. With one conversion triggered, the logic should return to its correct pre-conversion state—but cases exist where two conversion cycles are necessary before the ADC is certain to perform a valid conversion. Hence, a good general rule is to perform two "dummy" conversions after powerup before relying on the results. (It is also well to recall that some ADCs react badly to having a conversion triggered before the previous conversion is complete; when this happens, one or two "dummy" conversions may be needed to return the logic to a known state.)

If an ADC's external logic is arranged so that the end of the ADC "Busy" signal starts a delay which ends with the start of the next conversion, it is important to realize that if the converter powers up in the Busy state, the Busy signal may remain latched up until a conversion Start pulse has been received. In this case, such a system cannot self-start. If the Busy signal is always present on power-up the problem is almost certain to be recognized—and addressed—during the design of the system; but if the Busy signal is only occasionally present on power-up the system may latch unpredictably. As a rule, control signals to an ADC during start-up should not depend on the logical state of Busy. ◘

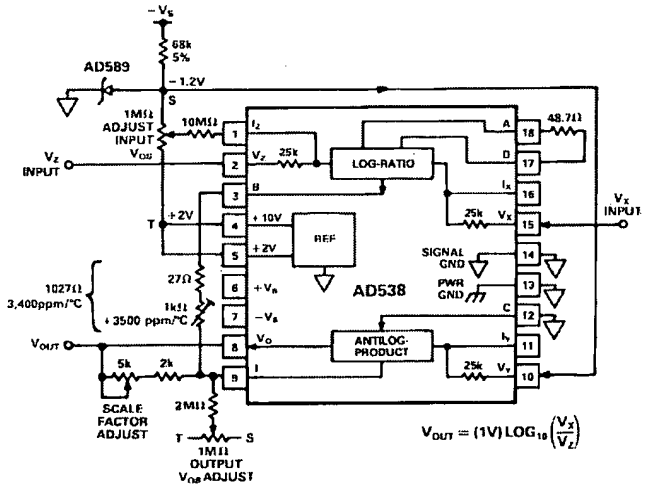
ABOUT LOG COMPENSATION RESISTORS

Q. Designs of logarithmic circuits, including those using the AD538 Y[Z/X]^m unit: (for example, Figure 6 from the AD538 Multifunction Unit data sheet†) call for "kT/q compensation resistors". What are they and where do I get them?*

A. The V_{BE} difference across two opposed silicon junctions, one carrying a current, I, and the other a current, I_{REF}, is (kT/q) ln (I/I_{REF}). Here, k/q is the ratio of Boltzmann's constant to the charge on an electron (about 1/11,605 K/V), and T is the absolute temperature in kelvins.

Although employing similar junctions in isothermal pairs eliminates the effects of temperature-sensitivity of reverse saturation current, the kT/q term is still temperature-dependent. To eliminate this dependency in the application, the logarithmic voltage must be used in a circuit whose gain is inversely proportional to the absolute temperature of the junctions. Over a reasonable range of temperatures near 20°C, this may be arranged by the use of a gain-setting 1-kΩ resistor having a positive temperature coefficient of approximately 3,400ppm/°C—and keeping it at the same temperature as the junctions.

A 3,500 ppm/°C resistor is available from Tel Laboratories, 154G Harvey Road, Londonderry, New Hampshire 03053 (603)-625-8994, Telex: (710)-220-1844, designated Q-81, and from the Precision Resistor Co. Inc., 10601, 75th. St., Largo, Florida 33543 [(813)-541-5771 Telex: 821788], as the PT146. Analog Devices offices in most European countries are aware of local suppliers of these resistors. ◘



*Much useful information about logarithmic and other analog function circuits can be found in the Nonlinear Circuits Handbook, published by Analog Devices ($5.95), P.O. Box 9902, Norwood MA 02062.

†See The Best of *Analog Dialogue* 1967–1991, pp. 164-167.

Ask The Applications Engineer—2

by James Bryant

WHEN IT COMES TO TRIMMING . . .

Q. I need some advice about trimming offsets and gains.

A. Don't!—unless you must. Good alternatives include (a) using headache-free devices, components, and circuits that meet the specs without trimming; (b) taking advantage of digital technology in system applications to make trim corrections in software. Savings provided on occasion by trim potentiometers, in conjunction with loosely spec'd devices, can turn out to be illusory when you consider the effects of circuit design, temperature, vibration, and life on performance and stability—as well as additional paperwork and complexity trimming entails.

Q. Nevertheless, how do I trim the offset and gain errors in analog circuitry?

A. In the correct order and with the correct inputs. If you consider the transfer characteristic of the circuit being trimmed the method to use is generally straightforward.

The simplified ideal transfer characteristic of a linear analog circuit (such as an amplifier, ADC or DAC) is given by the equation:

$$OP = K \times IP \quad (1)$$

where OP is output, IP is input, and K is a scale factor (Note that this simplification hides an enormous number of issues: quantization error in an ADC; dimensionality of K if the input and output are in different forms [e.g. voltage in / current out]; intentional offsets; and many others.)

In a real (non-ideal) circuit, offset and gain errors, OS (referred to the input) and ΔK, respectively, also appear in the equation, which becomes:

$$OP = (K + \Delta K) \times (IP + OS) \quad (2)$$

$$OP = (K \times IP) + [(K \times OS) + (\Delta K \times IP) + (\Delta K \times OS)] \quad (3)$$

Equations (2) and *(3)* are incomplete in that they assume only one offset—at the input—but this is the most-common case. Systems with separate input and output offsets will be considered later.

From *(3)* we see that it not possible to trim gain directly when an unknown offset is present. Offset must be trimmed first. With IP set at 0, the offset trim is adjusted until OP *is* also 0. Gain may then be trimmed: with an input near to full scale (FS), the gain trim is adjusted to make the output obey equation (1).

Q. But what about bipolar ADCs and DACs?

A. Many ADCs and DACs may be switched between unipolar and bipolar operation; such devices, wherever possible, should have their offset and gain trimmed in the unipolar mode. Where it is not possible, or where the converter is to operate only in the bipolar mode, other considerations apply.

A bipolar converter may be considered as a unipolar converter with a large offset (to be precise, an offset of 1 MSB—one-half of full-scale range). Depending on the architecture used, this bipolar offset *(BOS)* may or may not be affected by the gain trim. If it is so affected, equation (1) becomes:

$$OP = K \times (IP - BOS) \quad (4)$$

In this case offset is trimmed at analog zero, after which gain is trimmed near FS—positive or negative, but usually positive. This is normally the method used for DACs where the bipolar offset is within the DAC.

If the bipolar offset is not affected by the gain trim:

$$OP = K \times IP - BOS \quad (5)$$

Here offset is trimmed at FS negative and gain is trimmed at (or very near to—see below) FS positive. This method is used for most ADCs and for DACs where bipolar offset is obtained by the use of op amps and resistors external to the DAC.

Naturally, the method suggested on the data sheet should always be followed, but where a data sheet is unobtainable, in general, offset should be trimmed at analog zero for DACs and FS negative for ADCs—and near FS positive for both.

Q. Why do you keep saying "near" to full scale?

A. Amplifiers and DACs may be trimmed at zero and full scale. In the case of a DAC, all-1's—the largest digital input possible—should produce an output 1 LSB below "full scale," where "full scale" is considered as some constant times the reference; this follows since the output of a DAC is the normalized product of the reference and the digital input.

ADCs are not trimmed at zero and FS. The output of an ideal ADC is quantized, and the first output transition (from 00 . . . 00 to 00 . . . 01) takes place 1/2 LSB above the nominal value of all 0's. Thereafter transitions take place every 1-LSB increase in analog input until the final transition takes place 1 1/2 LSB below FS. A non-ideal ADC is trimmed by setting its input to the nominal value of a desired transition and then adjusting until the ADC output flickers between the two values equally.

The offset of an ADC is therefore trimmed with an input corresponding to the first transition (i.e., 1/2 LSB above zero or above FS negative—which is "near" zero or "near" FS negative); and the gain is then trimmed at the last transition (i.e. 1 1/2 LSB below FS positive—which is "near" FS positive). This procedure results in an interaction between the gain and offset errors during offset trim but it should be too slight to be significant.

Q. Are there any other anomalies resulting in a need to trim "near", rather than at full scale?

A. Synchronous voltage-to-frequency converters (SVFCs) are liable to injection locking phenomena when their output frequency is harmonically related to their clock frequency, i.e., when their output is very close to 1/2, 1/3 or 1/4 of clock frequency. FS for an SVFC is 1/2 clock frequency. The presence of a trim tool can exacerbate the problem. It is therefore advisable to trim the gain of an SVFC at around 95% of FS.

Q. What about circuits requiring both "input" and "output" offset trim?

A. Circuits such as instrumentation and isolation amplifiers often have two stages of dc gain, and the gain of the input stages can be variable. Thus a two stage amplifier, with an input offset, *IOS*, an output offset, *OOS*, a first stage gain of *K*, and a unity-gain output stage, has (for zero input) an output, *OP*, of:

$$OP = OOS + K \times IOS \qquad (6)$$

From (6) it is evident that if the gain is constant we need only adjust either *IOS* or *OOS* to null the total offset (although if the input uses a long-tailed pair of bipolar transistors we will get a better offset temperature coefficient if we trim both—for FET long-tailed pairs this is not necessarily the case). If the first stage gain is to be varied, both offsets must be trimmed.

This is done by an iterative procedure. With zero input, and gain set to maximum, the input offset is adjusted until the output is also zero. The gain is then reduced to its minimum value and the output offset adjusted until the output is zero again. The two steps are repeated until no further adjustment is necessary. Gain trimming should not be done until both *IOS* and *OOS* are pulled; the actual values of the high and low gains used in offset trim are unimportant.

Q. What circuitry should I use for gain and offset trims?

A. Many amplifiers (and a few converters) have terminals for trimming gain and offset. Many more do not.

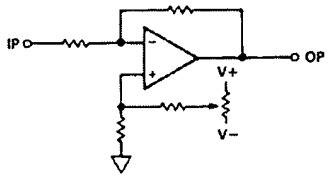
Offset trim is normally performed with a potentiometer connected between two assigned terminals, and its wiper is connected (sometimes via a resistor) to one of the supplies. The correct connections and component values will be given on the device data sheet. One of the commonest differences between op-amps is the value of offset correction potentiometer and which supply it should be connected to.

Where separate terminals are not provided for offset trim, an offset-adjusting constant can usually be added to the input signal. Two basic possibilities are shown in Figures 1a and 1b. Where the correction is being made to a system where a differential input op amp is used as an inverter (the commonest case) the method of Ia is best to correct for device offsets—but not system offsets. In the single-ended connection, method 1b will work for system offsets but should be avoided where possible for small device offsets, because it often requires a very large value of summing resistance, compared to the signal-input resistances, in order to (i) avoid loading the summing point excessively, (ii) scale the correction voltage properly and produce enough attenuation to minimize the effects of differential supply-voltage drifts. It is often helpful to use resistances between the supplies and the potentiometer to increase trim resolution and reduce dissipation.

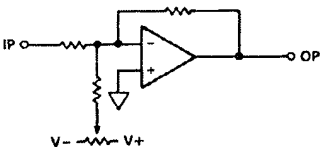
Where gain trim is provided for in a circuit, it will generally consist of a variable resistor. Details of its value and connection will appear on the data sheet of the device. Where gain trim is not required, this resistor may be replaced by a fixed resistor having half the resistance of the maximum value of the recommended trim potentiometer.

Where gain trim is not provided it is not always achievable externally without an additional variable-gain stage. For example, consider a DAC using a ladder network. If the ladder network is used in the current mode (Figure 2a), the input impedance at the reference terminal does not vary with digital code, and the gain of the DAC may be trimmed with a small variable resistor in series with either the reference input or the feedback resistor. However, if the DAC is used in the voltage mode (Fig 2b), then the reference input impedance is code dependent, and gain may only be trimmed by varying the reference voltage—which is not always possible—or the gain of the buffer amplifier.

The possibility of trimming gain in circuits not furnished with gain-trim circuitry, therefore, will depend on individual cases; each must be assessed on its own merits. ◘

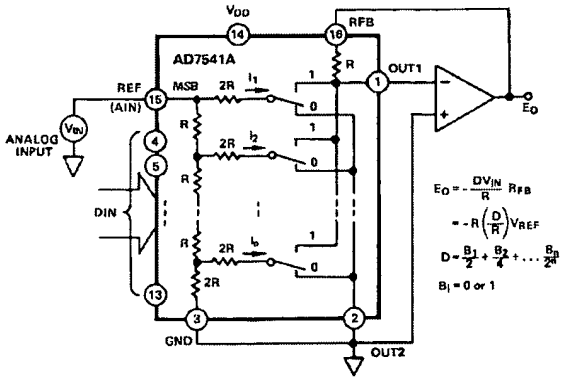


a. Voltage applied to non-inverting input.

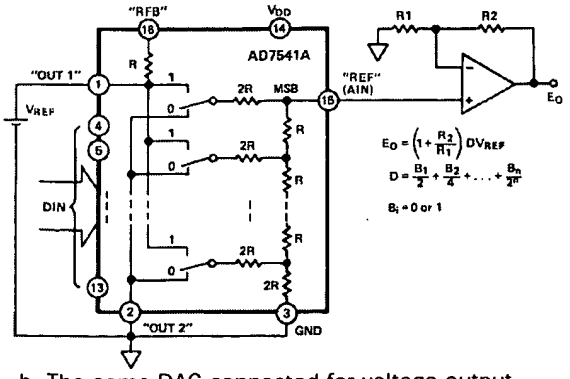


b. Current summed at inverting input.

Figure 1. Two connections for offset adjustment.



a. CMOS DAC connected for current steering. Input impedance is constant.



b. The same DAC connected for voltage output.

Figure 2. Comparing basic DAC circuits.

close to the upper transition, etc.). That may be sufficiently accurate for your application, but if you add noise to the input of the converter—so that several codes can appear at the output—you will find that the code distribution contains information to place the dc value of the input more exactly.

In the earlier examples involving the AD1879, we saw how the code distribution looks when the input is in the vicinity of a code transition; the two most-frequent output codes are the ones on either side of the transition. Their average is therefore a good estimate of where the input lies. In fact, taking the average of a lot of conversions, while the input stays put, is an excellent way of enhancing the resolution of the converter. One has to be careful, when processing the converter output, to allow the output word length to grow without introducing roundoff errors. Otherwise one actually injects unwanted noise—called *requantization noise*—into the final output. Note that filtering out the noise is only just that; it will have no effect on other error sources of the converter, such as integral and differential nonlinearity.

This concept of resolution enhancement is an interesting one and is not restricted to the dc domain. One can actually trade resolution for bandwidth in the ac domain and combine the outputs of several converters or to construct a more-accurate output. The basic principle is that signal repetitions (which are self-correlated) add linearly, while repetitions of random noise produce root-square increases. Thus, a fourfold increase in number of samples increases S/N by 6 dB.

Q: *You mentioned a couple of converter ac specifications above. I am somewhat confused about how S/N, THD+N, THD, S/THD, S/THD+N, and dynamic range are measured on A/D and D/A converters and how they relate to each other. Can you shed any light on this?*

A: Your confusion is quite understandable. There is unfortunately no industry standard on exactly how these quantities are measured and therefore, what exactly they mean. Sometimes manufacturers are guilty of choosing the definition that portrays their part favorably.

Most often data sheets include a note on the testing conditions and how the different specs were calculated. The best advice I can give is to read these very carefully. By simple calculations you can often convert a specification for one part to a number that allows a fair comparison to a specification for another part.

Most specifications are not expressed in absolute units, but as relative measurements or ratios. Noise, for example, is not specified in rms volts, but as *SNR*, or the ratio between signal power and noise power under particular test conditions. These ratios are usually expressed in decibels, dB, and occasionally as percentages (%). A power ratio, x, expressed in bels, is defined as $\log_{10}x$; multiply by 10 if expressed in decibels (one tenth of a bel): $10 \log_{10}x$. *SNR* is therefore equal to $10 \log_{10}$ (signal power/noise power) dB. Evaluated in terms of rms voltage quantities, $SNR = 20 \log_{10}(V_{signal}/V_{noise})$.

Armed with this knowledge, let's see whether we can make sense out of the multiple specifications you mentioned above (many of which are redundant). Those specifications seek to describe how the imperfections of the converter affect the characteristics of an ac signal that gets processed by the converter. For dc applications, a listing of the magnitude of the actual imperfections suffices, but these can only suggest ac performance. For example, integral nonlinearity is a major factor in determining large-signal distortion (along with glitch energy for D/As) while differential nonlinearity governs small-signal distortion. To accurately determine the ac performance, at least two types of tests are performed in the case of A/Ds. The tests are as follows:

a) Full-scale sine

A sinusoidal signal approaching full-scale is applied to the converter. The signal is large enough so that converter's imperfections cause significant harmonic components to occur at multiples of the input signal frequency. The harmonics will show up in the output spectrum, along with noise. A common performance measure is the relative magnitude of the harmonic components, usually expressed in dB. Relative to what? Two possibilities are the applied input signal and the full scale of the converter (which in most cases is different from the applied input signal). Referring the harmonics to full scale will clearly yield a lower (more attractive) number than referring them to the rms value of the actual input signal. This reference issue causes a lot of confusion when dynamic specifications are evaluated, because there is no universally accepted standard for what each performance measure should be referred to. The best advice I can give you is: never assume anything; read manufacturers' data sheets very carefully.

Sometimes the magnitudes of the individual harmonics are specified, but most often only the total harmonic distortion (THD) is specified. The THD measures the total power of the harmonics and is found by adding the individual harmonics in rss fashion. The formula then for *THD* when referred to the input signal is

$$20 \log_{10}\left[\frac{\sqrt{\sum_{i=2}^{m} H^2(i)_{rms}}}{S}\right] \text{ or } 10 \log_{10}\left[\frac{\sum_{i=2}^{m} H^2(i)_{rms}}{S^2}\right]$$

where $H(i)_{rms}$ refers to the rms value of ith harmonic component and S to the rms value of the input signal. Usually, harmonics 2 through 5 are sufficient. Note that the input-frequency, or *fundamental*, component is the first harmonic. To refer any harmonic to full scale, add x dB to the formula above, where x is the magnitude of the input signal relative to full scale. This simple conversion formula can be applied to other specifications, but take care to observe proper polarity of the log quantities.

Nowadays, clear distinction is usually made between total harmonic distortion plus noise (*THD+N*) and *THD*. This has not always been the case. *THD+N* includes not only the harmonics that are generated in the conversion, but also the noise. The formula for *THD+N* when referred to the input signal is:

$$20 \log_{10}\left[\frac{\sqrt{N^2_{rms} + \sum_{i=2}^{m} H^2(i)_{rms}}}{S}\right]$$

or

$$10 \log_{10}\left[\frac{N^2_{rms} + \sum_{i=2}^{m} H^2(i)_{rms}}{S^2}\right]$$

where N_{rms} is the rms value of the integrated noise in the bandwidth specified for the measurement.

Another commonly used specification is signal to noise-plus-distortion ($S/[N+D]$, or $S/[THD+N]$), also called *sinad*. This is essentially the inverse of $THD+N$, when referred to the signal; its dB number is the same, but with opposite polarity.

Another performance measure describing the test results is the signal to noise ratio, S/N or SNR, which is a measure of the relative noise power, most useful for estimating response to small signals in the absence of harmonics. If S/N is not specified, but THD and $THD+N$ are provided, relative to the input signal, THD can be rss-subtracted from $THD+N$ to obtain the noise to signal ratio [= $1/(S/N)$]. If the numbers are given in dB, the rss subtraction formula for logarithmic quantities in the Appendix can be used as follows

$$SNR = -10\log_{10}\left(10^{(THD+N)/10} - 10^{THD/10}\right)$$

to yield the input signal power relative to noise power expressed in dB.

b) Low-level sine

The second test usually performed is to apply a sinusoidal signal well below full scale to the converter (usually –60 dB). At this input level, sigma-delta converters usually exhibit negligible nonlinearities, so only noise (no harmonic components) appears in the spectrum. At this level, $S/N = S/N+D = -THD+N = -THD$, when all are referred to the same level. As a result, one specification indicating the noise level suffices to describe the result of this test. This specification called *dynamic range* (inversely, dynamic-range *distortion*), specifies the magnitude of the integrated noise (and harmonics if they exist) over a specific bandwidth relative to full scale, when a –60-dB input signal is applied to the converter.

Conventional (i.e. not sigma-delta) converters can exhibit harmonics in their output spectrum even with low-level input signals because all the codes may not have equal width (differential nonlinearity). In some such instances, the S/N, which ignores harmonics, measured with a –60-dB input signal, is different from dynamic range.

Frequently one sees $THD+N$ at –60-dB and dynamic range specified for the same converter. These really are, as explained above, redundant since they only differ in the reference used. The only twist on dynamic range is that sometimes, when audio converters are specified, a filter that mimics the frequency response of the human ear is applied to the converter output. This processing of the converter output is called A-weighting (because an A-weighting filter is used); it will effectively decrease the noise floor, and therefore increase the signal-to-noise ratio, if the noise is white.

Everything discussed above applies to both A/D and D/A converters, with the possible exception of signal to noise ratio. Sometimes (particularly for audio D/A converters) S/N is a measure of how "quiet" the D/A output is when zero (midscale) code is sent to the converter. Under these conditions, the S/N expresses the analog noise power at the D/A output relative to full scale output.

It's important to note that the performance measures above are affected by: *bandwidth* of the measurement, the *sampling frequency*, and the input *signal frequency*. For a fair comparison of two converters, one has to make sure that these test conditions are similar for both.

Image Filtering Question

Q: I intend to use Analog's AD1800 family of audio D/A converters for a digital audio playback application. I understand that using an interpolator ahead of the D/A will make it easier to filter the D/A output, assuming I want to get rid of all the images at the D/A output. But is it really necessary to filter the output, since all the images will be above the audible range as long as sampling is at >40 kHz?

A: Good question. The audio equipment (audio amplifiers, equalizers, power amplifiers, etc.) that may eventually receive the output of your D/As are typically built to handle 20-Hz to 20-kHz signals. Since they are not intended to respond at frequencies much beyond 20 kHz—and in effect themselves function as filters—they may not have the necessary slew rate and gain to handle incoming signals from an unfiltered D/A output having significant energy well above 20 kHz. With their slew-rate and gain limitations, the amplifiers are driven into nonlinear regions, generating distortion. These distortion products are not limited to high frequencies but can affect the 20-Hz to 20-kHz range as well. Attenuating the high frequency signals at the DAC will therefore reduce the possibility of distortion. CD players often include filters steep enough to reduce the total out-of-band energy to >80 dB below full scale.

APPENDIX

RSS addition of logarithmic quantities: The root-square sum of two rms signals, S_1 and S_2, has an rms value of $\sqrt{S_1^2+S_2^2}$. One often needs to calculate the rss sum of two numbers that are expressed in dB relative a given reference. To do this one has to take the antilogs, perform the rss addition, then convert the result back to dB. These three operations can be combined into one convenient formula: If D_1 and D_2 are ratios expressed in dB, their sum, expressed in dB, is

$$10\log_{10}\left(10^{D_1/10} + 10^{D_2/10}\right)$$

Similarly, to find the difference between two rms quantities,

$$x = \sqrt{S_2^2 - S_1^2}$$

the result, x, expressed in dB, is

$$10\log_{10}\left(10^{D_2/10} - 10^{D_1/10}\right)$$ ◘

References:

[1] *Oversampling Delta-Sigma Data Converters—Theory, Design, and Simulation*, edited by J.C. Candy and G.C. Temes, IEEE Press, Piscataway, NJ, 1991.

[2] J. Vanderkooy and S.P. Lipshitz, "Resolution Below the Least Significant Bit in Digital Systems with Dither," *J. Audio Eng. Soc.*, vol. 32, pp. 106-113 (1984 Mar.); correction ibid., p.889 (1984 Nov.).

[3] A.H. Bowker and G.J. Lieberman, *Engineering Statistics*, Prentice Hall, Englewood Cliffs, NJ, 1972.